From K-mean to Co-Clustering to Rich Community Discovering

Tri Nguyen [1]

Abstract

Tensor decomposition has been propular for decades because of its distinctive characteristics compared to matrix. There are many works dedicated to exploiting tensor potential through various areas, particularly in graph clustering. However, there is a lack of a clear connection between traditional clustering and tensor-based methods. Showing relation between these techniques under graph clustering problem would benefit for a deeper understanding as well as further directions to employ tensor techniques.

Moreover, tensor decomposition has many variants, such as Canonical Polyadic Decomposition (CPD), and LL1 decomposition. While both of them are all compelling, but blunt applying these techniques would not produce any effective result. This technical report aims to: firstly, linking tensor decomposition techniques to a very well-known k-mean clustering algorithm, and based on that, motivating the usage of LL1 decomposition over CPD in graph clustering problem.

1. Introduction

Graph data is prevailed in our life, especially under the current increasing popularity of social networks. Even though this structure introduces more challenging to mining tasks because of its high dimension and complex correlation compared to other data structures, data presented under graph structure encodes very rich information. Graph clustering (or graph partitioning) is one of the most common exploratory mining tasks that can unravel and reveal many insights into a network, especially when the network is large and complicated. In graph clustering, the intent is to partition the graph's nodes into clusters where inter-cluster connectivity is sparse and intra-cluster connectivity is dense.

Nodes within a cluster have many connections among them, which shows that they are more related to each other than the rest. Under this view, graph clustering could be classified as a general unsupervised cluster problem. Then, a natural question is whether k-mean, a very well-known algorithm in the family of unsupervised clustering algorithms, can solve graph clustering problem?

In the purest form of k-mean algorithm, only vector representation is acceptable. However, there were some works that come up with k-mean variants to work with graph data. For instance, (Lozanovska et al., 2004) bridged the relation between k-mean and spectral clustering by introducing weighted kernel k-mean objective function which is a more generalized version of the pure k-mean objective function. Since spectral clustering method can apply to graph data, the proposed algorithm can be utilized with either graph or vector input.

In this work, we are more interested in the pure k-mean version in terms of making a connection between k-mean with graph clustering. Particularly, instead of just changing the objective function, we seek a methodology that generalizes naturally and gradually from pure clustering method to graph clustering.

The overall of generalizing is depicted in Figure 1. Firstly, an immediate generalization of k-mean is co-clustering. Co-clustering was proposed in (Hartigan, 1972), and recently gained attraction in applications, including gene co-expression to network traffic and social network analysis (Charrad & Ben Ahmed, 2011). Co-clustering is more flexible than k-mean in the sense that k-mean imposes clustering on the whole column vectors while co-clustering allows clustering on selective elements of columns.

Co-clustering brings a natural generalization to arbitrary higher-way co-clustering (Papalexakis et al., 2013b). That means, it is inherently suitable to work on higher-order data, such as tensor. When the order is higher than 2, co-clustering coincides with a tensor decomposition technique, called Canonical Polyadic decomposition (CPD). The evolvement of using CPD is very natural which is described in detail later. Here, it is worth noting that CPD imposes a very simple assumption on data input: it assumes that each cluster can be represented by a rank-1 matrix, which in terms of graph clustering is a clique.

[1]Department of Electrical Engineering and Computer Science, Oregon State University, US. Correspondence to: Tri Nguyen <nguyetr9@oregonstate>.

Matrix analysis course, 2020 - Instructor: Prof. Xiao Fu, EECS, OSU

The assumption of CPD does not always hold in real data since there are other structures. The work (Koutra et al., 2015) provided a list of structures playing as vocabulary of a graph, including (near-)clique, star, chain, and bipartite. And that limitation motivates (Gujral et al., 2020) to come up with a final step of generalization: utilize LL1 decomposition instead of CPD, which enhance informativeness of decomposed factors by allow them to be up to rank $L > 1$.

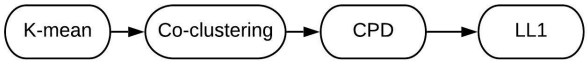


Figure 1. From K-mean to LL1

2. Tensor preliminaries

Tensor is basically an array indexed by n indices. It is a generalized notion of matrix. While matrix always has the dimension of 2 which uses 2 indices, tensor can have an arbitrary number of dimensions, also known as modes or ways. Hence, matrix is just 2-mode tensor. Most applications are interested in 3-mode tensor, albeit they apply to higher-order tensor. Therefore, for the rest of this report, we only use tensor as a 3-mode tensor. One simple way of interpreting tensor is using its analog to familiar matrix notation.

Definition 2.1 (Fibers, Slabs, Norm)**.** Fibers are analogue to rows and columns of matrix. Fibers are a collection of elements from $\mathcal{X}$ when fixing all dimensions except one, denoted as $\mathcal{X}(i, k, :)$. Fixing all dimensions except two gives us slice, denoted as $\mathcal{X}(i, :, :)$

Frobenius norm of tensor $\mathcal{X}$ is square root of sum of the squares of all elements. It is similar to Frobenius norm of matrix, e.i. $\|\mathcal{X}\|_F = \sqrt{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\mathcal{X}(i,j,k)^2}$

Definition 2.2 (Tensor product)**.** Any matrix rank F can be written as a summation of F outer products of 2 vectors: $\mathbf{X} = \sum_{f=1}^{F}\mathbf{a}_i\mathbf{b}_i^T$. Applying the same strategy for 3 vectors: a 3-mode tensor can be always written as a summation of F "outer products" of 3 vectors: $\mathcal{X} = \sum_{i=1}^{F}\mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i$. Here, $\circ$ denotes tensor product or outer product operator. Factor $\mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f$ forms a rank-1 tensor: $\mathcal{T} = \mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f \Leftrightarrow \mathcal{T}(i,j,k) = \mathbf{a}_f(i)\mathbf{b}_f(j)\mathbf{c}_f(k)$

Similar to matrix, if tensor $\mathcal{X}$ can be expressed as summation of F rank-1 tensors, then $\mathbf{X}$ has rank at most F.

Definition 2.3 (Kruskal rank)**.** The Kruskal rank $k_\mathbf{A}$ of matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is the largest integer k such that any k columns of $\mathbf{A}$ are linearly independent.

From the definition, it is clear that $k_\mathbf{A} \leq \text{rank}(\mathbf{A}) \leq \min(m, n)$

2.1. CP decomposition

Canonical polyadic decomposition (CPD) (Harshman, 1970), also known as Parallel factor analysis (PARAFAC) is a decomposition method that aims to factor tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ into sum of F rank-1 tensors

$$\mathcal{X} = \sum_{f=1}^{F}\mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f \tag{1}$$

where $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_F] \in \mathbb{R}^{I\times F}$, $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_F] \in \mathbb{R}^{J\times F}$, $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_F] \in \mathbb{R}^{K\times F}$. For simplicity, Eq.1 is shortened as $\mathcal{X} = [[\mathbf{A}, \mathbf{B}, \mathbf{C}]]$. One typical choice to find $\mathbf{A}, \mathbf{B}, \mathbf{C}$ is to treat the problem as an optimisation problem

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \left\| \mathcal{X} - \sum_{f=1}^{F}\mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f \right\| \tag{2}$$

Similar to matrix where SVD always decomposes any matrix into sum of rank-1 matrices, any tensor can be decomposed by CPD up to a sufficiently large F. Opposite to the matrix case, CPD for tensor comes with attractive characteristic of uniqueness under mild condition.

Theorem 2.1. *(Kruskal, 1977) Given $\mathcal{X} = [[\mathbf{A}, \mathbf{B}, \mathbf{C}]]$ with $\mathbf{A} \in \mathbb{R}^{I\times F}$, $\mathbf{B} \in \mathbb{R}^{K\times F}$, $\mathbf{C} \in \mathbb{R}^{K\times F}$. If $k_\mathbf{A} + k_\mathbf{B} + k_\mathbf{C} \geq 2F + 2$ and rank$(\mathcal{X}) = F$ then the decomposition of $\mathcal{X}$ is essentially unique.*

2.2. LL1 decomposition

A more generalized decomposition is LL1. It is a decomposition of a tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ rank R into sum of R blocks with rank-$(L_r, L_r, 1)$:

$$\mathcal{X} = \sum_{r=1}^{R}(\mathbf{A}_r\mathbf{B}^T) \circ \mathbf{c}_r$$

Again, LL1 can be solved as an optimisation problem:

$$\min_{\mathbf{A}_r,\mathbf{B}_r,\mathbf{c}} \left\| \mathcal{X} - \sum_{r=1}^{R}(\mathbf{A}_r\mathbf{B}^T) \circ \mathbf{c}_r \right\|_F^2$$

(Lathauwer, 2011) gave theorem of uniqueness of LL1 under some conditions.

Theorem 2.2. *Given $\mathbf{A} \in \mathbb{R}^{I\times LR}, \mathbf{B} \in \mathbb{R}^{I\times LR}, \mathbf{C}^{K\times R}$ are LL1 decomposition of $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$. Also assume that $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are drawn from certain absolutely continuous distributions. If $\lfloor \frac{IJ}{L^2} \rfloor \geq R$ and*

$$\min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{J}{L} \right\rfloor, R\right) + \min(K, R) \geq 2R+2$$

then $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are almost surely unique up to scaling and permutation ambiguities.

3. From K-mean to Co-clustering

Unsupervised clustering plays an important role in many discovery and mining applications. In its pure form, unsupervised clustering aims to group entire column vectors into clusters, which enforces these column vectors to appear similar or be closed respect to certain metrics, such as Euclidean distance.

In this section, we examine k-mean, which is a well-known algorithm in this class. The relationship between k-mean and co-clustering was explained explicitly in the work of (Papalexakis et al., 2013b). A similar approach was also provided by (Wu et al., 2017), although the authors did not mention co-clustering. Suppose we have to cluster J data points $\{\mathbf{x}_j \in \mathbb{R}^I\}_{j=1}^J$ into R clusters ($R \ll J$), the problem can be formulated as an optimization problem

$$\min_{\mathbf{M},\mathbf{B}\in\mathcal{RS}} \left\|\mathbf{X} - \mathbf{M}\mathbf{B}^T\right\|_F^2 \tag{3}$$

where

- $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_J] \in \mathbb{R}^{I\times J}$ is input data
- $\mathbf{M} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_R] \in \mathbb{R}^{I\times R}$, $\boldsymbol{\mu}_i$ is i-th cluster centers
- $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_R] \in \mathbb{R}^{I\times R}$ is assignment matrix, each i-th element in $\mathbf{b}_r$ represents how much i-th element involving in r-th cluster
- $\mathcal{RS}$ is row sum-to-one constraint $\sum_{r=1}^R \mathbf{B}(j,r) = 1$. It ensures every data point must belong to some clusters

Note that $\mathbf{A}\mathbf{B}^T$ can be written as a summation of R rank-1 matrices, then Eq.3 becomes

$$\min_{\boldsymbol{\mu}_i;\mathbf{b}_i\in\{0,1\}^{J\times K}\cap\mathcal{RS}} \left\|\mathbf{X} - \sum_{i=1}^K \boldsymbol{\mu}_i\mathbf{b}_i^T\right\|_F^2 \tag{4}$$

Although the minimization problem contains 2 variables $\mathbf{M}$ and $\mathbf{B}$, applications might only concern about matrix $\mathbf{B}$ since it is directly related to the physical meaning of the result. Some constraints on this matrix might vary dependent upon assumptions of cluster membership. Hard clustering is applied if $\mathbf{B}$ as a binary matrix. Oppositely, soft clustering is considered when the binary constraint is relaxed to non-negativity.

Another variance called 'lossy' clustering is originated from relaxing $\mathcal{RS}$ constraint to $\sum_{k=1}^K \mathbf{B}(j,k) \leq 1$. It is also equivalent to dropping $\mathcal{RS}$ constraint, since $\mathbf{B}(j,:)$ could be always normalized. In this case, magnitude of $\mathbf{B}(i,j) \geq 0$ indicates how strong data points $\mathbf{x}_i$ belonging to cluster j. Under this relaxation, it is possible to have some data points not belong to any cluster. If there are outliers in the data, it is reasonable to not consider these anomaly data points to resulting clusters. An additional constraint of sparsity can be imposed on $\mathbf{B}$ to emphasize that most data points would not belong to a given cluster.

Considering all these variants, k-mean clustering imposes no constraints on $\boldsymbol{\mu}$. From this perspective, it is natural to think of K-mean as a one-sided clustering involving only one mode of data matrix $\mathbf{X}$. Then one can implement the same treatment of $\mathbf{B}$ on matrix $\mathbf{M}$ to obtain two-sided clustering or namely *co-clustering*. The idea applies to higher dimension. Based on 'soft lossy' variance of k-mean, and similar to Eq.4, co-clustering is modeled as following

$$\min_{\substack{0\leq\mathbf{a}_i\leq 1,\\ 0\leq\mathbf{b}_i\leq 1}} \left\|\mathbf{X} - \sum_{i=1}^R \mathbf{a}_r\mathbf{b}_r^T\right\|_F^2$$

where '> 0' should be interpreted element-wise, and both $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_r], \mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_r]$ should contain many zeros.

One of the options to impose sparsity is adding l_1 penalty to objective function

$$\min_{\substack{0\leq\rho_i\leq\bar{\rho},\\ 0\leq\mathbf{a}_i,\mathbf{b}_i\leq 1}} \left\|\mathbf{X} - \sum_{i=1}^K \rho_i\mathbf{a}_i\mathbf{b}_i^T\right\|_F^2 + \lambda_a\sum_{i,k}|\mathbf{A}(i,k)| + \lambda_b\sum_{j,k}|\mathbf{B}(j,k)| \tag{5}$$

Here, λ_a, λ_b play as sparsity penalty weights on row and column, respectively. Note that we need both λ_a, λ_b here instead of λ because of the possibly different sparseness in different modes of data matrix $\mathbf{X}$. Similar to K-mean, the magnitude of $\mathbf{B}(i_1,j)$, $\mathbf{A}(i_2,j)$ indicates the strongness of node type 1 i and node type 2 j regarding cluster j.

4. From co-clustering to LL1 decomposition for richer structure

Co-clustering could be easily applied to higher order $\mathbf{X}$, e.g. third order tensor $\mathcal{X}$. In this case, objective function in Eq.5 becomes:

$$\min_{\substack{0\leq\rho_r\leq\bar{\rho}\\ 0\leq\mathbf{a}_r,\mathbf{b}_r,\mathbf{c}_r\leq 1}} \left\|\mathcal{X} - \sum_{f=1}^R \rho_r\,\mathbf{a}_r\circ\mathbf{b}_r\circ\mathbf{c}_r\right\|_F^2 + \lambda_a\sum_r^K\|\mathbf{a}_r\|_1 + \lambda_b\sum_r^K\|\mathbf{b}_r\|_1 + \lambda_c\sum_r^K\|\mathbf{c}_r\|_1 \tag{6}$$

The first component is CPD decomposition problem, where $\mathcal{X}$ is assumed to be a summation of R rank-1 tensors.

In the context of graph clustering, one can formulate $\mathcal{X}$ such that each slice of $\mathcal{X}$ is a snapshot of an adjacency matrix or a different aspect of a multi-aspect graph. The objective function above results each co-cluster represented by a rank-1 tensor $\mathbf{a}_r\circ\mathbf{b}_r\circ\mathbf{c}_r$, where i-th entry in $\mathbf{a}_r$ implies how well node i fits in cluster r respect to mode 1, and so on for $\mathbf{b}_r, \mathbf{c}_r$.

Then, $\mathbf{c}_r$ (since it is temporal axis), and use $\mathbf{a}_i \circ \mathbf{b}_i = \mathbf{a}_i \mathbf{b}_i^T$ to extract clusters as the same as a co-clustering on matrix.

Uniqueness of CPD under mild condition is an attractive feature compared to its counterpart of matrix. However, CPD implicitly assumes that the cluster's structure could be modeled by rank-1 tensors. Since rank-1 tensor does not have much 'freedom', clusters described by these tensors limits to a very simple structure. Indeed, the only structure fits this simplistic modeling is clique. Similarly to co-clustering for matrix, $\mathbf{a}_i, \mathbf{b}_i$ are assignment matrices corresponding to mode-1, mode-2 of tensor $\mathcal{X}$, respectively. It lacks extra-cluster connectivity, which can be seen as the tradeoff of this simplistic model is to lose structure data of the cluster. Therefore, CPD implicitly enforces a uniform connection among nodes within the cluster. Every node with similar roles is perfectly described by a (near-)clique.

A general graph frequently contains many richer structures than cliques. Enhancing flexibility in encoding graph structure is essential in graph clustering. The authors of (Gujral et al., 2020) proposed LL1 as a replacement of CPD to archive that goal.

LL1 model applying to graph clustering is formulated as

$$\{\mathbf{A}, \mathbf{B}, \mathbf{C}\} \leftarrow \underset{\mathbf{A},\mathbf{B},\mathbf{C}}{\arg\min} \left\| \mathcal{X} - \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^T) \circ \mathbf{c}_r \right\|_F^2 + r(\mathbf{A}) + r(\mathbf{B}) + r(\mathbf{C})$$

where:

- $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is data tensor.
- $\mathbf{A}_r \in \mathbb{R}^{I \times L_r}$ is full column rank, $\mathbf{B}_r \in \mathbb{R}^{J \times L_r}$ is full column rank
- Matrix $\mathbf{A}$ is constructed by stacking R matrices: $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_R] \in \mathbb{R}^{I \times LR}$
- Matrix $\mathbf{B}$ is constructed by stacking R matrices: $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_R] \in \mathbb{R}^{J \times LR}$
- $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_R] \in \mathbb{R}^{K \times R}$
- $r(.)$ represents contraint function. In (Gujral et al., 2020), author applied 2 constraints: non-negative and sparsity which are also accordant with (Papalexakis et al., 2013b)

Similar to co-clustering using CPD, each pair of $\mathbf{A}_r, \mathbf{B}_r$ forms a cluster. The most important advantage of LL1 is that instead of describing cluster by rank-1, $\mathbf{A}_r \mathbf{B}_r^T$ forming a matrix rank L_r to model cluster, which allows encoding more sophisticated structures.

5. Experimental results

In this section, we run some experiments to demonstrate the effectivity of CPD and its generalization LL1 under the view of graph clustering. All the experiments use synthetic data to have full control that makes it easier to reveal the superior characteristic of LL1 over CPD.

5.1. General setup

The setup described in this sub-section is applied to all experiments. The detailed descriptions for each experiment are explained later.

According to (Koutra et al., 2015), a graph can be divided into primitive structures, such as clique, star, chain, that can ease graph discovery tasks. Hence, we will have experiments with these structures.

A tensor $\mathcal{X} \in \mathbb{R}^{50 \times 50 \times 10}$ encodes an undirected weighted graph either involving by time or a certain view of the graph. We generate $R = 5$ clusters, each includes 10 nodes. A frontal slab $\mathcal{X}(:, :, i)$ represents an adjacency matrix corresponding to view i. To visualize the input graph in 2D as Figure 2, tensor $\mathcal{X}$ is undergone mean aggregation across the third mode.

The frontal $\mathcal{X}(:, :, i)$ is constructed particularly to form a certain type of network, such as clique, star, or chain. All weights within each cluster are drawn from the uniform distribution. To make the problem more realistic, noise are introduced: a uniform distribution is added to all weighted connections. The role of noise is twofold: it is an intrinsic factor of any real problem, and it provides natural inter-cluster connectivity.

The third mode is sampled from the uniform distribution. This choice is motivated by multiple views of a graph. Although clusters are fixed, their appearances might be different from each other under different viewpoints.

With these settings, the following experiments are only different terms of constructing each frontal slab $\mathcal{X}(:, :, i)$

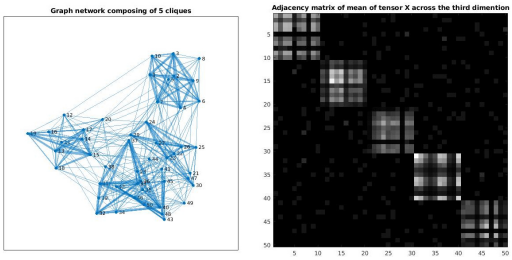
5.2. Co-clustering for clique discovery

As can be seen in graph visualization Figure 2a and its adjacency matrix Figure 2b, there are 5 clique clusters with different densities.

Strong connections are depicted by thicker edges in 2a and lighter cells in 2b, and opposite for weak connections. Although all clusters are clique, each cluster has its weight distribution which makes their inner pattern different from each other.

Clique is the simplest structure since membership assignment is the only information needed to describe a clique. That means the co-clusters model described in Eq.6 perfectly fits the problem. In particular, each cluster can be fully described by either $\mathbf{a}$ or $\mathbf{b}^T$, since they are all membership assignment, and should be mostly identical in case of an undirected graph (Papalexakis et al., 2013a). Then, each

cluster now can be formed by a rank-1 matrix $\mathbf{Q} = \mathbf{a}\mathbf{b}^T$, where $\mathbf{a}, \mathbf{b}$ are drawn randomly. Note that tensor $\mathcal{X}$ satisfies the uniqueness condition in Theorem 2.1.



(a) Graph visualisation (b) Adjacency matrix

Figure 2. Input data: a graph composing of 5 clique clusters. (a) Thicker edges are corresponding to higher value of weights. (b) Lighter cells are corresponding to higher values

Running CPD with $R = 5$ reveals correctly all 5 clusters and their clique structures, as it appears in Figure 3. Especially, the inner pattern of each cluster is also reconstructed adequately compared to the original input data. It demonstrates CPD capacity of clustering and reconstructing intra-cluster connectivity. This efficiency of CPD is attributed to its uniqueness under mild condition.

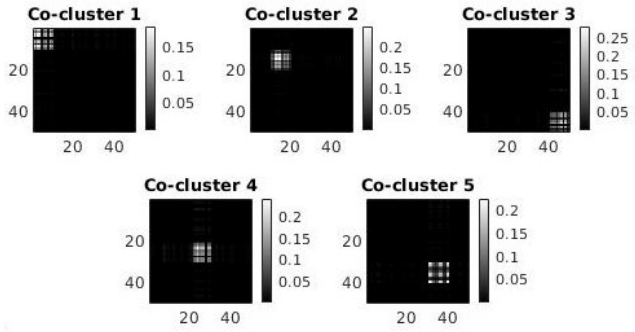


Figure 3. Co-clustering based on CPD

5.3. LL1 for richer structure

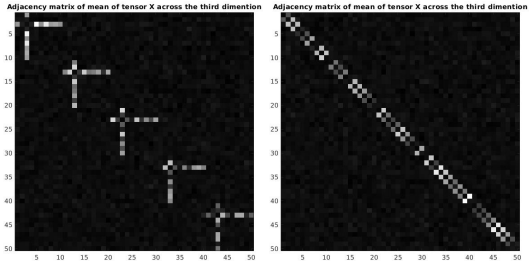
Similarly to the previous part, Figure 4a and 4b are adjacency matrices of graphs comprising of 5 star clusters and 5 chain clusters, respectively. Unlike clique, representation of star/chain requires specified node-to-node interactions within each cluster. It is infeasible for a rank-1 matrix to fully describe the cluster. Specifically, a cluster of n nodes needs a matrix at least rank-2 to describe star structure, and rank-n to describe chain structure. It can be expound by examining the adjacency matrix of these structures without noise.

In case of adjacency matrix $\mathbf{Q}$, all rows are 0 expect row i, and all columns are 0 except column j. Denote | and — as columns and rows which are different from zeros, then rank of matrix can be derived as:

$$\text{rank}(\mathbf{Q}_{\text{star}}) = \text{rank}\left(\left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & 0 \end{array}\right]\right) = \text{rank}\left(\left[\begin{array}{|c|} \hline 0 \\ \hline \end{array}\right]\right) = 2$$

$$\text{rank}(\mathbf{Q}_{\text{chain}}) = \text{rank}\begin{pmatrix} 0 & q_{12} & 0 & 0 & \dots & 0 \\ q_{21} & 0 & q_{23} & 0 & \dots & 0 \\ 0 & q_{32} & 0 & q_{34} & \dots & 0 \\ 0 & 0 & q_{43} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & q_{n-1} & 0 \end{pmatrix}$$

Adjacency matrix of chain structure $\mathbf{Q}_{\text{chain}}$ is a tridiagonal matrix with bandwidth of 1 and rank of $n - 1$.



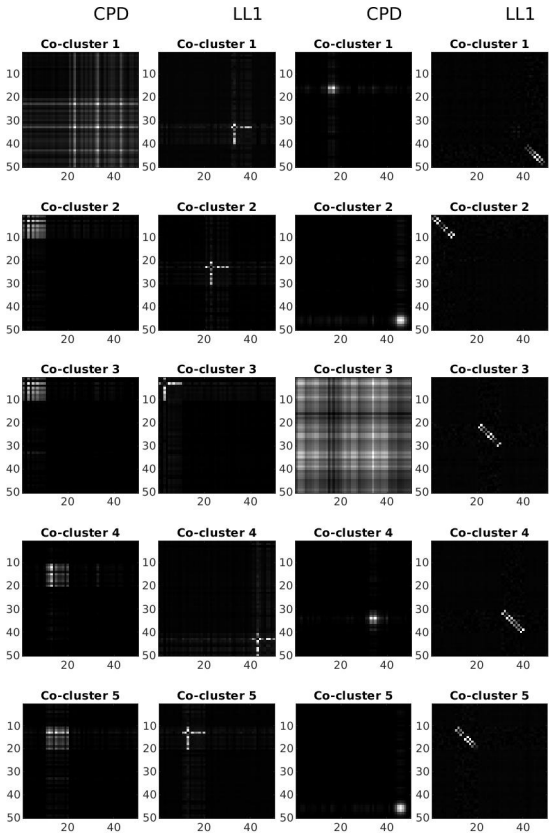
(a) A graph composing of 5 star clusters (b) A graph composing of 5 chain clusters

Figure 4. Adjacency matricies of graphs composing of richer structure communities

Performance co-clustering using CPD and LL1 on 2 graphs with $R = 5$ results as Figure 5. LL1 surpasses significantly CPD in both graphs. The failure of CPD on discovery clusters as well as reconstructing the inner pattern of each cluster is because of its oversimplified assumption about rank-1 describing matrix. This assumption is relaxed and extended by LL1 which brings effective results. Although LL1 model is more flexible than LL1, it still possesses uniqueness features under a mild condition as described in Theorem 2.2. And thanks to that features, all 5 clusters in Figure 5b, Figure 5d are recovered perfectly in terms of both membership assignment and inner structure.

6. Conclusion

In this work, we have demonstrated the fruitfulness and necessity of generalization of clustering to co-cluster. Then it is again essential to establish a further generalization of co-cluster under CPD to LL1 decomposition. The experiments using synthetic data have shown limitation of CPD and how LL1 overcomes it by extending rank-1 to rank-L. The result shows that LL1's assumption works effectively on graph clustering with rich structures components.



(a) CPD performs on star graph (b) LL1 performs on star graph (c) CPD performs on chain graph (d) LL1 performs on chain graph

Figure 5. Superior performance of LL1 compares to CPD

References

Charrad, M. and Ben Ahmed, M. Simultaneous Clustering: A Survey. In Kuznetsov, S. O., Mandal, D. P., Kundu, M. K., and Pal, S. K. (eds.), *Pattern Recognit. Mach. Intell.*, pp. 370–375, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21786-9.

Gujral, E., Pasricha, R., and Papalexakis, E. Beyond Rank-1: Discovering Rich Community Structure in Multi-Aspect Graphs. In *Proc. Web Conf. 2020*, number Mdl in WWW '20, pp. 452–462, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380129. URL https://doi.org/10.1145/3366423.3380129.

Harshman, R. a. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Work. Pap. Phonetics*, 16(10): 1– 84, 1970. URL http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf.

Hartigan, J. A. Direct Clustering of a Data Matrix. *J. Am. Stat. Assoc.*, 67(337):123–129, 1972. ISSN 1537274X. doi: 10.1080/01621459.1972.10481214. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1972.10481214.

Koutra, D., Kang, U., Vreeken, J., and Faloutsos, C. Summarizing and understanding large graphs. *Stat. Anal. Data Min. ASA Data Sci. J.*, 8(3):183–202, 2015. doi: 10.1002/sam.11267. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11267.

Kruskal, J. B. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.*, 18(2):95–138, 1977. ISSN 00243795. doi: 10.1016/0024-3795(77)90069-6.

Lathauwer, L. D. E. DECOMPOSITIONS OF A HIGHER-ORDER TENSOR IN BLOCK TERMS—PART II: DEFINITIONS AND UNIQUENESS*. *SIAM J. Matrix Anal. Appl.*, 30(3):1033–1066, 2011.

Lozanovska, M., Dhillon, I. S., Guan, Y., and Kulis, B. Kernel K-Means: Spectral Clustering and Normalized Cuts. In *Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, volume 28 of *KDD '04*, pp. 551–556, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138881. doi: 10.1145/1014052.1014118. URL https://doi.org/10.1145/1014052.1014118.

Papalexakis, E. E., Akoglu, L., and Ience, D. Do more views of a graph help? Community detection and clustering in multi-graphs. *Proc. 16th Int. Conf. Inf. Fusion, FUSION 2013*, pp. 899–905, 2013a.

Papalexakis, E. E., Sidiropoulos, N. D., and Bro, R. From K-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors. *IEEE Trans. Signal Process.*, 61(2):493–506, 2013b. ISSN 1053587X. doi: 10.1109/TSP.2012.2225052.

Wu, J., Wang, Z., Wu, Y., Liu, L., Deng, S., and Huang, H. A Tensor CP Decomposition Method for Clustering Heterogeneous Information Networks via Stochastic Gradient Descent Algorithms. *Sci. Program.*, 2017, 2017. ISSN 10589244. doi: 10.1155/2017/2803091.